

# CS 420-01 Software Engineering

## Description and Objectives

Prerequisite: CSCI 223

An introduction to current techniques used in medium and large-scale software development. Topics include requirements analysis, functional specification, systems design, implementation, testing, maintenance, project management, and professional ethics.

This course introduces you to the terminology, methods, activities and artifacts used in the development of medium and large enterprise-quality software projects. Coding skill is only a small part of what makes software projects succeed. More often than not, software projects fail for ethical, professional or organizational reasons, rather than technical risk. Therefore, it is important to examine those reasons, and to practice discipline-specific team problem-solving and communication skills.

Part of the practice is to begin working on a team project that has the potential to continue on into the capstone Software Practicum (CSCI 421) or a Senior Research (CSCI 499) courses. If you do an internship, either for credit (CSCI 491) or not for credit, you will find that this course either helps prepare you to succeed, or rounds out ideas and practices you may already have encountered.

### Course Sections

Section 01

MWF 10:00-10:50, 216 Thompson Hall

### Text

Essentials of Software Engineering, 3<sup>rd</sup> Edition  
Frank Tsui, Orlando Karam, Barbara Bernal  
ISBN: 978-1-4496-9199-8  
Publisher: Jones & Bartlett, 2014

### Reference Books

- [Producing Open Source Software](#), Karl Fogel
- Any reasonable guide to technical writing

### Course Website

<http://macs.citadel.edu/rudolphg/csci420/f13>

### Tools/Materials

- A tool for drawing UML diagrams
- An IDE for writing/managing code (such as Eclipse)

### Instructor Information

|              |                                                                                                   |
|--------------|---------------------------------------------------------------------------------------------------|
| Instructor   | George Rudolph                                                                                    |
| E-mail       | <a href="mailto:george.rudolph@citadel.edu">george.rudolph@citadel.edu</a>                        |
| Office       | 225 Thompson Hall                                                                                 |
| Phone        | 953-5032                                                                                          |
| Office Hours | M 9-10am, W 1-2pm, R 3-5pm; by appointment; when my door is open; email is a good way to reach me |

# Course Requirements

This course involves reading, writing and in-class activities. You are expected to read and absorb the text and other materials on your own outside of class. You are expected to know all assigned material even if it is not discussed in class. You will write the several technical engineering documents over the course of the term. This writing should be taken seriously. You will need to put in consistent effort throughout the whole semester. You will need to speak up in class, answer questions, and present ideas orally and in writing.

## Attendance & Participation

Attendance is mandatory and expected. You should be punctual, and have read the assigned material before class. In order to properly pace your way through unit work and project deliverables, there will be something due nearly every class period—one or two unit questions or a project document—either individually or as a team. Reviews of project documents will be conducted in class, and it is important for all team members to be present on those days.

In general, late submissions will not be accepted, and there is no extra credit work. Exceptions may be made at the instructor's discretion.

## Units and Unit Sheets

Below is a list of the 7 major units we will be covering this semester. The topics and readings cover all of the chapters in the text. The pacing is roughly one chapter per week of coverage, with two chapters per unit.

Each Unit has a corresponding Unit Sheet that lists assignments and point values. Think of each unit sheet as a combination of traditional homework assignments and take-home exam. Unit sheets should be downloaded and printed by each student. The instructor can sign off items that the student has completed on the sheet, providing a backup record of work.

Questions are grouped in layers. Each unit has a "C" layer, a "B" layer, and an "A" layer. "C" layer is

designed to help you learn basic and general information. You will choose "C" layer questions you want to answer, complete the work, and then demonstrate learning by oral defense. Oral defense means you have to answer questions about your work asked by the instructor. "B" is designed to help you apply, information from the "C" layer, discover something, hypothesize or problem solve. Typically, choose one question to work on in the "B" layer. The "A" layer requires you to analyze an issue, research facts and form an opinion—critical analysis. Typically, choose one question from the "A" layer. Rubrics for each kind of question will be available so that you know how answers will be graded. Each 5 points on a question represents about 20 minutes worth of work.

## Team Project

The team project is to be completed as a group. The ideal project is a real-world application which you identify and begin working on in this course, and potentially carry on into a capstone course—but continuing is not required. The project work is represented by 8 documents as follows: Project Plan, User's Manual, Software Requirements, Test Plan, Test Report, Software Architecture, Software Design, Acceptance Test.

More details for each document will be discussed in class, especially once you have chosen a project. Your textbook contains outlines and templates for a number of the documents you must write. *Use them!*

Each member of the team is responsible to know and understand documents that another team member has written. Every document should be reviewed by at least one other member of the team *before* it is turned in for review and grading by the instructor.

## Final Exam

The final exam is comprehensive, and will be offered at the scheduled date and time.

## GRADING

I do not “take off” points for errors; you earn points for appropriate work.

### ATTENDANCE & PARTICIPATION 100 PTS

Silent participation in class will not earn full credit. Sleeping or texting will reduce your grade.

### UNITS 700 PTS

7 Units, 100 points each.

### PROJECT 200 PTS

8 Project Documents as scheduled.

### Final Exam 200 PTS

Date: Sat., Dec. 14, 0800, TH 216

### TOTAL 1200 PTS

## Grading Scale

**A 1080+**

**B 960+**

**C 840+**

**D 720+**

**F 719-0**

## Tentative Schedule

| Units & Readings                                                                                               | Important Dates                                                                           |
|----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Unit 1: Introduction <u>Unit Sheet</u><br>Reading: Chapter 1, 3<br>Dates: Aug 28 – Sep 11                      | 9/2 Labor Day: Class Held<br>9/4 Project Plan Due<br>9/11 Unit 1 Due                      |
| Unit 2: Modeling the Process <u>Unit Sheet</u><br>Chapters 4, 5<br>Dates: Sep 13 - 27                          | 9/27 Unit 2 Due                                                                           |
| Unit 3: Requirements <u>Unit Sheet</u><br>Chapter 2, 6<br>Dates: Sep 30 – Oct 14                               | 10/4 Parent’s Day: Class Held<br>10/14 Unit 3 Due                                         |
| Unit 4: Architecture & Design <u>Unit Sheet</u><br>Chapters 7,8<br>Dates: Oct 16 - 30                          | 10/23 Leadership Day: No Class<br>10/30 Unit 4 Due                                        |
| Unit 5: Implementation, Testing and Quality Assurance <u>Unit Sheet</u><br>Chapters 9, 10<br>Dates: Nov 1 - 15 | 11/15 Unit 5 Due                                                                          |
| Unit 6: Builds, Support and Maintenance <u>Unit Sheet</u><br>Chapter 11, 12<br>Dates: 18 - 22                  | 11/22 Unit 6 Due; Class Held                                                              |
| <i>Fall Break</i>                                                                                              | <i>Nov 25 -29</i>                                                                         |
| Unit 7:Secure Software and Project Management <u>Unit Sheet</u><br>Chapter 13, 14<br>Dates: Dec 2 - 11         | 12/11 Unit 7 Due, Last Day of Class<br>12/11 Project Due<br><b>12/14 Final Exam, 0800</b> |

# **Policies and Advice**

---

**What should I be able to know and do after this course (Student Learning Outcomes)?**

Explain the significance of the various products produced during application design

Explain the various processes used to manage a software design project

Have some experience with the development of a software project

Discuss the ethics of a variety of professional situations

now how to read and report (oral and written) on Technical material

---

**What is the course policy on technology use in class?**

Follow the College's policy on use of cellphones and electronic devices in class. In general, no electronic devices should be seen or heard in class, unless explicitly permitted by the instructor. You may bring a laptop to class, but use should be confined to note-taking or accessing course-related materials. Any other uses will reduce your participation grade.

---

**What if I need disability accommodations?**

If you have a documented disability, please see me at the beginning of the course so we can plan how to help you be successful.

---

**What is the course policy on plagiarism, academic honesty and academic integrity?**

Do your own work. You must demonstrate academic integrity when taking exams, writing papers. You **must cite** sources of information or ideas that are not your own. This includes clips from books or the Internet. You cannot turn in a paper someone else has written and claim it is your own work. Plagiarism is an automatic 0 for any exam or assignment, may cause you to fail the course at the instructor's discretion, and may incur further action by the college. Please know and follow the Honor Code and The Citadel's Academic Integrity Policy.

---

**What if I am feeling overwhelmed by this course?**

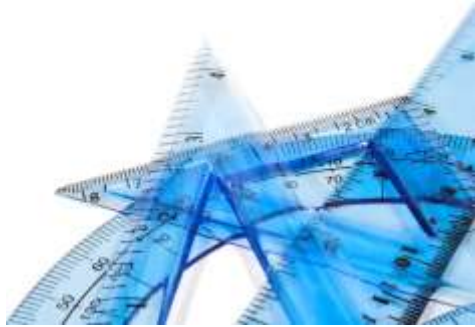
Ask questions in class. Most likely other students have the same questions. Come see the instructor during office hours or drop in.

---

**What other resources might help me succeed in this course?**

Talk to the Science Librarian in the Daniel Library. Visit the Academic Support Center. Both resources can help with any stage of the writing process.

---



## RUBRICS

### How do I do...

There are two kinds of rubrics here, common rubrics for assignments that could be used across many courses (such as flash cards), and rubrics that are specific to software engineering work products. General rubrics are taken from the book ["Layered Curriculum" by Kathie F. Nunley](#). The others are my creation but follow in the same spirit. The point is to clarify grading expectations for the student, and to ease grading for the instructor.

### Flash Cards

You may either buy index cards or make flash cards from regular white 8.5" x 11" white paper. Fold one sheet of paper in half three times to make 8 squares and cut along the fold lines. Use 2 or 3 pieces of paper, enough for the number of terms. Write the word on one side, and the definition in your own words on the other. Learn them. They are worth 10 points.

Grading: I will choose 10 cards at random and ask you about those words. You get 1 point for each question you get correct.

### Bookwork

Book work involves reading a section of a textbook and answering questions at the end of the reading. You may write the answers down, or simply learn them in your head.

Grading: I will ask you 3 questions at random from the reading. You will get 5 points for each question you answer correctly.

### Artwork

Artwork involves posters and models. Generally, they are worth 10 points. Be creative. Artistic ability counts in the grade. Note: Mock-up User interfaces, and engineering diagrams are in a different category than artwork.

Grading: I will ask you 5 things you learned about the project. Points are based on artistic value and learning.

### Miscellaneous Reading

You may read for 45 minutes on a course-related topic in newspapers, magazines, books, etc. Please tell me beforehand what topic you are reading on. Upon completion, I will ask you to tell me about your reading.

Grading: Your summary is worth 10 points based on enthusiasm and information gained from the reading. You should be able to explain and defend 5 things you learned from the reading.

### Labs

Labs include any "B" level questions, including what we traditionally think of as labs and any question in which you must design a procedure to solve your problem. Choose one of the following ways to present your results:

1. Lab report  
A written summary that includes what question you are trying to answer, your hypothesis, details of the procedure you used, what happened and your conclusions.
2. Lab Display  
Using a piece of paper or a computer model, draw or illustrate what happened in your lab. Make sure the question is on display and that we can see the results and your conclusions.
3. Lab Verbal Report  
Prepare a 3-5 minute report containing the information as specified for a Lab Report above. Give the report in person, or record it on video and post the video on your website or youtube, or somewhere accessible to me.

## **"A" Level Assignments**

Choose an "A" level question. Go to the library (or use a scientific search engine) and find at least 3 recent journal or magazine articles on your topic. Recent means less than 5 years old.

Your write-up has two main parts. First, summarize the main points of each article. Second, write a good 2- or 3-paragraph summary of your opinion on the topic. As a rule of thumb, a good paragraph contains 5-7 sentences, and has a clear main point. Make sure you cite your research when stating your opinion. Get help from a librarian on finding good sources if you need it. Grammar and spelling matter. These assignments are worth 20 points.

## **General Comments about Papers, etc.**

When doing "A" level assignments, essays and papers, follow Chicago Manual of Style with numbered references when questions arise that are not specifically answered or understood from your instructions.

## **Watching Videos**

Videos are generally worth 15 points for each 45-60 minutes you watch. To get points, you must pay attention and watch during the required minutes. If you are using sites such as youtube, this may require that you watch a number of videos on a topic. 10 points are given for watching, and up to 5 points for you answering 5 questions I ask about what you've learned.

## **Worksheets**

"C" level worksheets are worth 5 points each, and you can do up to three on any unit for which they are available, for 15 points total.

Grading "C" level: I will ask you 3 questions about the worksheet topic. Points are based on your responses to the questions I ask.

"B" level worksheets, when available, have fewer problems than "C" level worksheets, but include more difficult questions requiring problem-solving. These are worth 15 points each.

Grading "A" level worksheets: I will ask you 3-5 questions about the worksheet topic. Points are based on your responses to the questions I ask.

## Lectures

Lectures are worth 5 points per day. You must be present for the entire lecture to get credit. An outline for the lecture may or may not be available for you to use to take notes on. If not, create your own outline and fill it in based on what I say in the lecture.

## Thank-you notes

The internet is full of guidelines for good thank-you notes. Thank you notes are worth 5pts.

## Project Plan

This document shows a set of major milestones and project deliverables and planned completion dates for each. The dates will likely change as you go along. That is OK as long as the original plan is reasonable. Project plan is worth 20 pts.

Grading: The document will be reviewed at least once by the instructor, and possibly by a customer, for changes. The final version will be given points.

## User's Manual

A User's Manual (UM) describes exactly how a user is to use the system for a given operation. In theory, the UM should be written from requirements, *before* the system is designed and implemented. The idea is to ensure that the system meets requirements as predefined, to avoid adding functions that do not need to be there. In practice, the UM is often left to the end of a project. The UM is worth 20 pts.

Grading: The document will be reviewed at least once by the instructor, and possibly by a customer, for changes. The final version will be given points. A good first cut at the system is what we are looking for.

## Software Requirements Document

The requirements document describes what the system is supposed to do, but usually not how it is supposed to do it. This is a technical engineering document for the team, and may also be part of a legal contract, but not for public consumption.

A requirement statement should be Specific, Measurable, Achievable, Reasonable and Testable. Requirements should be numbered or labelled using a consistent scheme that is easy to learn, use, maintain, understand and reference. Requirements should be grouped together in logical groupings. The set of requirements should be reasonably Complete and Consistent. The Requirements document is worth 30 pts.

Grading: The document will be reviewed at least once by the instructor, and possibly by a customer, for changes. The final version will be given points. A good first cut is what we are looking for.

## Test Plan

A good test plan is your plan for how system testing is going to be conducted. This includes staffing, equipment and tools, a basic schedule, pass/fail criteria, test environment(s), and any other factors that might impact testing the system that is being designed. The Test Plan is worth 20 pts.

Grading: The document will be reviewed at least once by the instructor, and possibly by a customer, for changes. The final version will be given points. A good first cut is what we are looking for.

## Test Report

A test report is the collection of all your system test cases. Good test cases have setup instructions, well-defined, well-ordered steps with observable results and clear pass/fail criteria. Each test cases references one or more requirements in the requirements document. The Test Report is worth 30 pts.

Grading: The document will be reviewed at least once by the instructor, and possibly by a customer, for changes. The final version will be given points. A good first cut is what we are looking for.

## **Software Architecture Document**

A software architecture document records the set of all design decisions you make about a system that affect the structure or organization of that system. Some people think of this as "high-level design". This document may also tools (like compilers), notations (like UML) and any other things that have a significant impact on the design or implementation of a system. The document does not include details that are not architecturally significant. The architecture document worth 30 pts.

Grading: The document will be reviewed at least once by the instructor, and possibly by a customer, for changes. The final version will be given points. A good first cut is what we are looking for.

## **Design Document**

A design document records the set of all design decisions and definitions that are important or useful, but not necessarily architecturally important. Some people call this "low-level design" because it fleshes out details between the architecture and implementation. This document should contain enough information that with it and the architecture document, a team could implement the system according to the design. The design document worth 30 pts.

Grading: The document will be reviewed at least once by the instructor, and possibly by a customer, for changes. The final version will be given points. A good first cut is what we are looking for.

## **Code Patch**

Doing a code patch generally requires that you download code from a repository, compile and run it, make changes, compile and tests your changes to show that they work, and upload your changes back to the repository.

The goal is to gain experience using some form of source code management tool, while doing a bit of round-trip coding and testing. The patch does not have to be complex. The patch may be a new feature, not limited to just bug fixes. The code patch is worth 20 pts.

Grading: You get 10 pts for showing the source code and demonstrating it working, and 10 pts for answering three questions about the code patch process.